IBM Docket No. AUS920010980US1

1

TITLE OF THE INVENTION

Method and System for Probing in a Network Environment

CROSS-REFERENCES TO RELATED APPLICATIONS

5   The present application is related to a co-pending application entitled Method and System for Performance Reporting in a Network Environment, filed on even date herewith, assigned to the assignee of the present application, and herein incorporated by reference.

10   FIELD OF THE INVENTION

The present invention relates generally to information handling, and more particularly to methods and systems for evaluating the performance of information handling in a network environment.

15   BACKGROUND OF THE INVENTION

Various approaches have been proposed for monitoring, simulating, or testing web sites. Examples include U.S. Pat. No. 6,278,966 B1 (Howard, et al., Aug. 21, 2001), "Method and System for Emulating Web Site Traffic to Identify Web Site Usage Patterns." However,

20   this example addresses substantially different problems (problems of simulation and hypothetical phenomena), and thus is significantly different from the present invention. Other examples include U.S. Pat. No. 6,078,956 (Bryant, et al., Jun.

25   20, 2000) and U.S. Pat. No. 5,787,254 (Maddalozzo, et al., Jul.28, 1998). Other examples include services available from vendors such as Atesto Technologies Inc., Keynote Systems, and Mercury Interactive Corporation. These services may involve a script that runs on a probe computer. The examples mentioned

30   above do not necessarily allow some useful comparisons.

A wide variety of valuable services are provided through client-server applications, so proper performance of client-server applications may be very important. Thus there is a need for systems and methods that evaluate the performance of client-server applications, including but not limited to web sites.

SUMMARY OF THE INVENTION

An example of a solution to problems mentioned above comprises providing a script; employing a plurality of probes, including at least one local probe and at least one remote probe; and measuring a client-server application's performance, with said probes, according to said script.

To give further examples of the solutions provided, there are solutions suitable for measuring any client-server application's performance. Also provided are solutions suitable for working applications that are actually providing services for users; the solutions do not depend upon mere simulations, and do not depend upon merely testing an application before it is put to work, for example. Also provided are solutions that allow comparing data from a local probe with data from a remote probe, for example.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained when the following detailed description is considered in conjunction with the following drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

FIG. 1 illustrates a simplified example of a computer system capable of performing the present invention.

FIG. 2 is a high - level block diagram illustrating an example of a measurement architecture employing a plurality of probes, according to the teachings of the present invention.

FIG. 3 is a block diagram illustrating an example of application measurement with the present invention.

FIG. 4 is a block diagram illustrating one example of how the present invention was implemented for web site measurement.

FIG. 5 is a diagram with flow charts illustrating selected features that may be included in an example of the present invention.

DETAILED DESCRIPTION

The examples that follow involve the use of one or more computers and one or more communications networks. The present invention is not limited as to the type of computer on which it runs, and not limited as to the type of network used.

The following are definitions of terms used in the description of the present invention and in the claims:

"Client-server application" means any application involving a client that utilizes a service, and a server that provides a service. Examples of such a service include but are not limited to: information services, transactional services, access to databases, and access to audio or video content.

"Comparing" means bringing together for the purpose of finding any likeness or difference, including a quantitative likeness or difference. "Comparing" may involve answering questions including but not limited to: "Is a measured response time greater than a threshold response time?" Or "Is a response time measured by a remote probe significantly greater than a response time measured by a local probe?"

4

"Computer-usable medium" means any carrier wave, signal or transmission facility for communication with computers, and any kind of computer memory, such as floppy disks, hard disks, Random Access Memory (RAM), Read Only Memory (ROM), CD-ROM, flash ROM, non-volatile ROM, and non-volatile memory.

"Measuring" means evaluating or quantifying.

"Performance" means execution or doing; "performance" may refer to any aspect of an application's operation, including availability, response time, time to complete batch processing or other aspects.

"Probe" means any computer used in evaluating, investigating, or quantifying performance; for example a "probe" may be a personal computer executing a script, acting as a client, and requesting services from a server.

"Response time" means elapsed time in responding to a request or signal.

"Script" means any program used in evaluating, investigating, or quantifying performance; for example a script may cause a computer to send requests or signals according to a transaction scenario. A script may be written in a scripting language such as Perl or some other programming language.

"Storing" data or information, using a computer, means placing the data or information, for any length of time, in any kind of computer memory, such as floppy disks, hard disks, Random Access Memory (RAM), Read Only Memory (ROM), CD-ROM, flash ROM, non-volatile ROM, and non-volatile memory.

FIG. 1 illustrates a simplified example of an information handling system that may be used to practice the present invention. The invention may be implemented on a variety of hardware platforms, including embedded systems, personal computers, workstations, servers, and mainframes. The computer

system of FIG. 1 has at least one processor 110. Processor 110 is interconnected via system bus 112 to random access memory (RAM) 116, read only memory (ROM) 114, and input/output (I/O) adapter 118 for connecting peripheral devices such as disk unit 120 and tape drive 140 to bus 112. The system has user interface adapter 122 for connecting keyboard 124, mouse 126, or other user interface devices such as audio output device 166 and audio input device 168 to bus 112. The system has communication adapter 134 for connecting the information handling system to a data processing network 150, and display adapter 136 for connecting bus 112 to display device 138. Communication adapter 134 may link the system depicted in FIG. 1 with hundreds or even thousands of similar systems, or other devices, such as remote printers, remote servers, or remote storage units. The system depicted in FIG. 1 may be linked to both local area networks (sometimes referred to as intranets) and wide area networks, such as the Internet.

While the computer system described in FIG. 1 is capable of executing the processes described herein, this computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the processes described herein.

FIG. 2 is a high - level block diagram illustrating an example of a measurement architecture employing a plurality of probes, according to the teachings of the present invention. The basic function of a probe is that it repeatedly executes a predefined scenario or script. As seen in FIG. 2, probes may be placed strategically in a hosting site or data center (at 211 or 212), at a local site outside the data center (at 271 or 272), on an enterprise network (intranet, at 280) or the Internet (at 290),

depending on the measurements that they need to provide. The example involves placing at least one remote probe, shown at 235, on the Internet, shown at 290. The list at 239 has descriptions of different measurement techniques that may be implemented with one or more probes placed at 235. The Internet measurements may be provided by various companies in the marketplace which provide this kind of service, or may be obtained by other means.

The example also involves placing at least one probe, shown at 221 and 222, in at least one data center, shown at 211 and 212. The list at 229 has descriptions of different measurement techniques that may be implemented with one or more probes placed at 221 or 222.

FIG. 2 illustrates that employing a plurality of probes may comprise at least one of: employing a component probe; employing an application probe; and employing a network probe. Referring to the lists at 229 and 239:

Component Probes measure availability, utilization and performance of infrastructure components, including servers, LAN, and services. Local component probes (LCPs) are deployed locally in service delivery centers or data centers (at 211 or 212).

Network Probes measure network infrastructure response time and availability. Remote Network Probes (RNPs) will be deployed in data centers (at 211 or 212) if measuring the intranet or at Internet Service Provider (ISP) sites if measuring the Internet.

Application Probes measure availability and performance of applications and business processes.

Local Application Probe (LAP): Application probes deployed in a local hosting site or data center (at 211 or 212) are termed Local Application Probes.

Remote Application Probe (RAP): An application probe deployed
from a remote location is termed a Remote Application Probe.

The concept of "probe" is a logical one. Thus, implementing a
local component probe could actually consist of implementing
multiple physical probes.

FIG. 3 is a block diagram illustrating an example of application
measurement with the present invention. This comprises: providing
a script (not shown in this diagram); employing a plurality of
probes, including at least one local probe (shown at 321, 322,
and 323) and one remote probe (shown at 331, 332, 333, 334, and
335); measuring a client-server application's performance
(applications shown at 301, 302, and 303) with said probes,
according to said script; and collecting in a database data
produced by said measuring (not shown in this diagram). Measuring
a client-server application's performance comprises measuring
response time for at least one request (the double-headed arrows
connecting probes and applications symbolize requests and
responses). Employing a plurality of probes may involve placing
at least one remote probe (such as those shown at 331, 332, and
333) on an intranet (shown at 380), or placing at least one
remote probe (such as those shown at 334 and 335) on the Internet
(shown at 390).

For example, providing a script would comprise defining a set of
transactions that are frequently performed by end users, and
employing a plurality of probes would comprise placing at least
one remote probe at each location having a relatively large
population of end users. Note that the Remote Application Probe
transactions and Local Application Probe transactions should be
the same transactions. The model measures all the transactions

locally, so that the local application response time can be compared to the remote application response time. This can provide insight to key application performance issues. End-to-end measurement of an organization's internal applications for internal customers will involve a RAP on the intranet, whereas end-to-end measurement of an organization's external applications for customers, business partners, suppliers, etc. will involve a RAP on the Internet. The model involves defining a representative transaction set, and deploying remote application probes at all relevant end-user locations. (This simplicity is something that can only be appreciated when this architecture is contrasted with other more complicated models.) A benefit following from the simplicity of this model is that it is easily generalized to other environments besides web based applications.

FIG. 4 is a block diagram illustrating one example of how the present invention was implemented for web site measurement. As an overview, this implementation comprised: providing a script (not shown here); obtaining at least one local probe measurement (local probe shown at 221) of a client-server application's performance (application shown at 301), according to said script; obtaining at least one remote probe measurement (remote probes shown at 235) of said client-server application 301's performance, according to said script; comparing at least one of said measurements (stored in database 422) with at least one threshold value (deriving said at least one threshold value from a service level agreement [SLA] shown at 462); and reporting results (for example, report 442) of said comparing.

Turning now to some details of the example implementation, obtaining at least one remote probe measurement (remote probes shown at 235) comprised measuring response time for a request

(the double-headed arrow connecting remote probes at 235 with application 301 symbolizes requests and responses). Obtaining at least one local probe measurement (local probe shown at 221) comprised measuring response time for said request (the double-headed arrow connecting local probe 221 with application 301 symbolizes requests and responses). The example implementation involved comparing at least one local probe measurement (in report 441 for example) with at least one remote probe measurement (in report 442 for example). The double-headed arrow at 450 symbolizes comparison.

Turning now to further details of the example implementation, we located application probes locally at hosting sites (local probe shown at 221, within data center 311) and remotely at relevant end-user sites (remote probes at 235). This not only exercised the application code and application hosting site infrastructure, but also probed the ability of the application and network to deliver data from the application hosting site to the remote end-user sites. End-to-end measurement of IBM external applications (symbolized by application 301 with web pages 401) for customers or business partners, for example, involved RAP's on the Internet (remote probes at 235 shown within Internet 290). While we measured the user availability and performance from a customer perspective (remote probes at 235), we also measured the availability and performance of the application at the location where it was deployed (local probe shown at 221, within data center 311). This provided baseline performance measurement data, that could be used for analyzing the performance measurements from the remote probes (at 235).

Local probe 221 was implemented with a personal computer, utilizing IBM's Enterprise Probe Platform technology, but other

kinds of hardware and software could be used. A local probe 221 was placed on the IBM network just outside the firewall at the center where the web site was hosted. A local probe 221 was used to probe one specific site per probe. There could be multiple

5   scripts per site. A local probe 221 executed the script every 20 minutes. Intervals of other lengths also could be used. If a local probe 221 encountered a problem (e.g. it was unable to access the site or unable to complete the script) on two consecutive executions of the script, local probe 221 generated a

10   real time alert (problem event), and sent it to a TIVOLI management system (shown as console 405). Another similar kind of management system could be used. An alert message via email also could be used.

15   Local probe 221 sent to a database 421 the data produced by the measuring process. Database 421 was implemented by using IBM's DB2 technology, but other database management software could be used, such as ORACLE, INFORMIX, SYBASE, MYSQL, Microsoft Corporation's SQL SERVER, or similar software. For local probe

20   data, an automated reporting tool (shown as report generator 431) ran continuously at set intervals, obtained data from database 421, and sent reports 441 via email to these IBM entities: the web site owner, the hosting center, and IBM's world wide command center. Reports 441 also could be posted on a web site at the set

25   intervals. Report generator 431 was implemented by using the Perl scripting language and the AIX operating system. However, some other programming language could be used, and another operating system could be used, such as LINUX, or another form of UNIX, or some version of Microsoft Corporation's WINDOWS, or some other

30   operating system.

Remote probes at 235 were implemented by contracting for probing services available from Mercury Interactive Corporation, but services from another vendor could be used, or remote probes could be implemented by other means (e.g. directly placing probes at various ISP's). A remote probe 235 may be used to probe one specific site per probe; a probe also has the capability of probing multiple sites. There could be multiple scripts per site. Remote probes 235 were located at various ISP's in parts of the world that the web site supported. A remote probe 235 executed the script every 60 minutes. Intervals of other lengths also could be used. If multiple remote probes at 235 are used, probe execution times may be staggered over the hour to ensure that the performance of the web site is being measured throughout the hour. Remote probes at 235 sent to a database 422 the data produced by the measuring process. Database 422 was implemented by using Mercury Interactive's database, but other database management software could be used, such as IBM's DB2, ORACLE, INFORMIX, SYBASE, MYSQL, Microsoft Corporation's SQL SERVER, or similar software. Report generator 432 was implemented by using Mercury Interactive's software and web site, but another automated reporting tool could be used, such as the one described above for local probe data (shown as report generator 431). IBM's arrangement with Mercury Interactive included the following: Mercury Interactive's software at 432 used IBM's specifications (symbolized by "SLA specs" at 462) and created near-real-time reports (symbolized by report 442) in a format required by IBM; IBM's specifications and format were protected by a confidential disclosure agreement; the reports at 442 were supplied in a secure manner via Mercury Interactive's web site at 432; access to the reports was restricted to IBM entities (the web site owner, the hosting center, and IBM's world wide command center).

12

FIG. 5 is a diagram with flow charts illustrating selected features that may be included in an example of the present invention. As an overview, this example comprises: providing a script (shown at 510); employing a plurality of probes, including at least one local probe (shown at 221) and at least one remote probe (shown at 235); providing an iterative process including a - d below:

a. measuring (shown at 521 and 531) a client-server application's performance (application shown at 301) according to said script 510;

b. sending (shown at 522 and 532) to a database (shown at 508) data produced by said measuring;

c. waiting for a set interval (shown at 523 and 533);

d. repeating (shown at 524 and 534) the above three steps until said iterative process is terminated;

with said at least one local probe 221, executing said iterative process (shown at 520); and with said at least one remote probe 235, executing said iterative process (shown at 530).

Turning now to some details of the example shown in FIG. 5, application 301 may be any client-server application. Some examples are a web site, a web application, database management software, a customer relationship management system, an enterprise resource planning system, or an opportunity-management business process where a client directly connects to a server.

Providing a script 510 may comprise defining a set of transactions that are frequently performed by end users. Providing a script 510 may involve decomposing an actual business process. The business process may for example: represent the most common tasks performed by the end users, exercise major

components of the applications, cover multiple hosting sites, cross multiple applications, or involve specific infrastructure components that should be monitored on a component level. Using script 510, local and remote application probes may measure

5   the end-to-end user experience for repeatable transactions, either simple or complex. End-to-end measurements focus on measuring the business process (as defined by a repeatable sequence of events) from the end user's perspective. End-to-end measurements tend to cross multiple applications, services, and

10   infrastructure. Examples would include: create an order, query an order, etc.

Measuring (521 and 531) a client-server application's performance typically would comprise measuring response time for at least one request. The double-headed arrow 501 connecting local probe 221

15   with application 301 symbolizes requests and responses. The double-headed arrow 505 connecting remote probe 235 with application 301 symbolizes requests and responses, sent via network 150. Requests are contained in script 510 that runs on

20   local probe 221 and remote probe 235. Employing a plurality of probes may comprise placing at least one remote probe on an intranet, or placing at least one remote probe on the Internet.

Some possible ways to implement local probe 221, remote probe

25   235, and one or more databases at 508 are described above regarding FIG. 4. Ways to implement a script that runs on a probe are well-known; vendors provide various services that involve a script that runs on a probe.

30   In conclusion, we have shown examples of methods and systems for probing client-server applications in a network environment.

One of the possible implementations of the invention is an application, namely a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of a computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, in a hard disk drive, or in a removable memory such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may be implemented as a computer-usable medium having computer-executable instructions for use in a computer. In addition, although the various methods described are conveniently implemented in a general-purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

While the invention has been shown and described with reference to particular embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and detail may be made therein without departing from the spirit and scope of the invention. The appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those with skill in the art that if a specific number of an introduced claim element is intended, such intent will be explicitly recited in the claim, and in the absence of such recitation no such limitation is present. For non-limiting example, as an aid to understanding, the appended claims may contain the introductory phrases "at

least one" or "one or more" to introduce claim elements. However, the use of such phrases should not be construed to imply that the introduction of a claim element by indefinite articles such as "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "at least one" or "one or more" and indefinite articles such as "a" or "an;" the same holds true for the use in the claims of definite articles.